DoDAF-Based Information Assurance Architectures

Dr. John A. Hamilton Jr.
Auburn University

The Department of Defense (DoD) Architecture Framework (DoDAF) is the prescribed means for documenting information systems in the DoD and is an integral part of the Joint Capabilities Integration and Development System. The inclusion of DoDAF architectures in new system development is mandated in DoD acquisition regulations and is resource-intensive. Deriving information assurance architecture from DoDAF-compliant architecture is a relevant way to leverage the mandatory investment in DoDAF architectures. Every software engineer supporting the DoD should be aware of the increasing importance of information assurance and the need for holistic approaches to security. Information assurance architectures described in this article offer a verifiable holistic approach to security.

A logical extension of the Department of Defense (DoD) Architecture Framework (DoDAF) is to specify and describe information assurance architecture. Such architecture, while primarily built on the DoDAF systems views (SVs), can also be supported by technical standards views and validated by operational views (OVs). Defense software engineers need to be aware of the DoDAF, the mandated system architecture in the Defense Acquisition System [1].

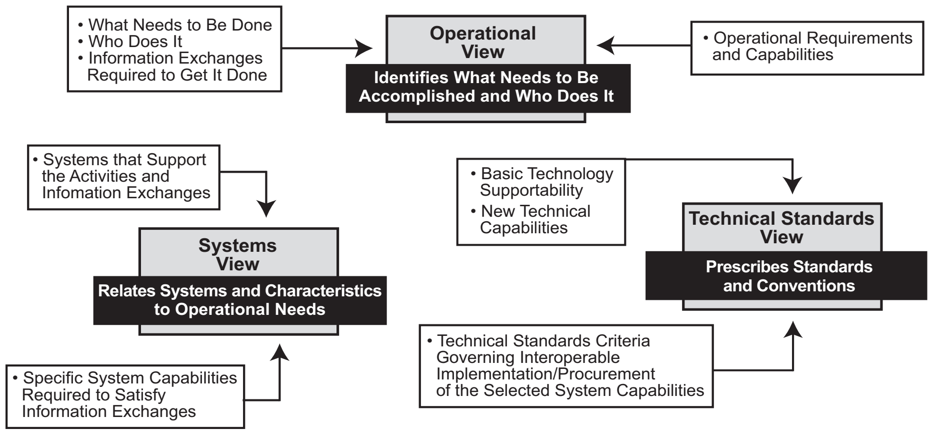
The information assurance architecture diagrams are primarily derived from the system architecture. At the Department of Computer Science and Software Engineering at Auburn University, we use low-level architecture work products to document the major security components and the application mechanisms and their interrelationships. Successful information assurance strategies require holistic solutions, i.e., architectures are valid and internally consistent, so it is logical to leverage the mandated DoDAF architectures for the basis of information assurance architecture.

We look at information assurance architecture to support network analysis and design to mitigate distributed denial-of-service attacks on bandwidth. We document open ports and required services to support a systematic software vulnerability analysis. Finally, we use the OVs to perform a requirements analysis to validate our architecture. We translate operational requirements into a modified, but DoDAF-compliant Systems Interface Description (SV-1) product and a Systems Communications Description (SV-2) product. We then use those products as a basis for constructing the rest of the information assurance architecture.

> *"A simple and effective rule for security design is the principle of least privilege. That is, allow only the minimum essential connectivity and functionality."*

We then validate the information assurance architecture against the system requirements and verify it against security regulations and instructions. This article will describe how to construct DoDAF-compliant information assurance architecture based on the research efforts of Auburn University and the practical application of that research.

Figure 1: *Relationship Between the DoDAF View*


• What Needs to Be Done
• Who Does It
• Information Exchanges Required to Get It Done
Operational View
Identifies What Needs to Be Accomplished and Who Does It
• Operational Requirements and Capabilities
• Systems that Support the Activities and Infomation Exchanges
• Basic Technology Supportability
• New Technical Capabilities
Systems View
Relates Systems and Characteristics to Operational Needs
Technical Standards View
Prescribes Standards and Conventions
• Technical Standards Criteria Governing Interoperable Implementation/Procurement of the Selected System Capabilities
• Specific System Capabilities Required to Satisfy Information Exchanges

Overview of the DoDAF

This author is somewhat skeptical of mandates in general [2]. However, the requirement for DoDAF architecture is mandated by both the Defense Acquisition System and several chairmen. Joint Chiefs of Staff instructions, most notably the Joint Capabilities Integration and Development System [3], make it seem likely that the DoD soon will have a critical mass of DoDAF-compliant architectures. It makes sense to leverage those architectures beyond satisfying acquisition requirements.

The mandatory use of the DoDAF is prescribed in DoD Instruction 5000.2, in which the Joint Staff is the assigned proponent for OVs, while the under secretary of defense (Acquisition, Technology, and Logistics), leads the development of the SVs in collaboration with the services, agencies, and combatant commanders [1]. Volumes I and II of the DoDAF, plus the DoDAF Deskbook, total more than 1,500 pages of documentation; the description that follows is necessarily abbreviated.

As defined in the DoDAF [4], an OV is "a description of the tasks and activities, operational elements, and information exchange required to accomplish DoD missions." An SV is "a set of graphical and textual products that describes systems and interconnections providing for, or supporting, DoD functions. The SV associates systems resources to OV." These relationships are outlined in Figure 1.

Further discussion of these three views is available online in "Modeling Command and Control Interoperability: Cutting the

Gordian Knot” [5].

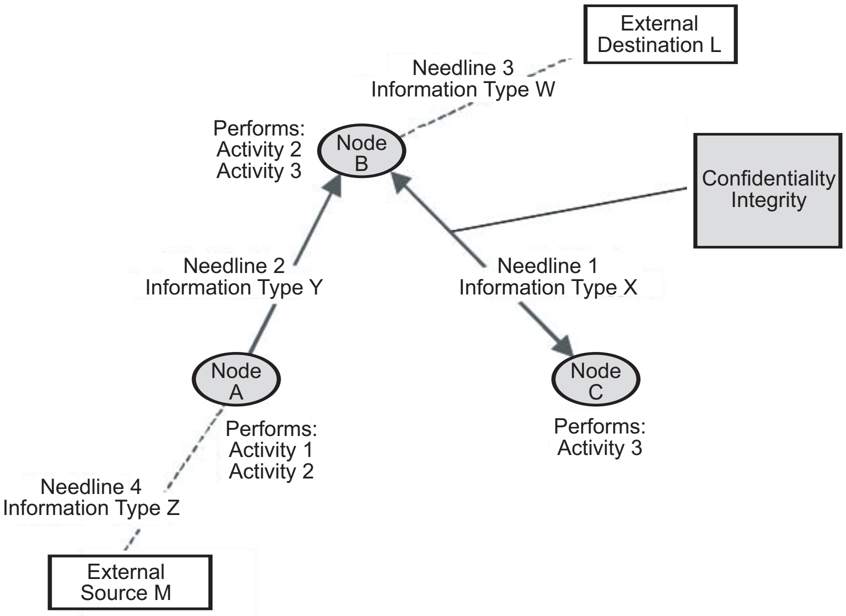

External Destination L
Needline 3 Information Type W
Performs: Activity 2 Activity 3
Node B
Confidentiality Integrity
Needline 2 Information Type Y
Needline 1 Information Type X
Node A
Node C
Performs: Activity 1 Activity 2
Performs: Activity 3
Needline 4 Information Type Z
External Source M

Figure 2: *DoDAF Deskbook Example of OV-2 With Security Attributes*

The technical standards view is essentially a listing of standards implemented by the systems in the architecture, and is now based on the DoD Information Technology Standards Registry found at <https://disronline.disa.mil>. Each communications system/network-enabled computer system defined in the SV will have an entry in the technical standards view outlining each standard used in the system.

The intellectual power of the DoDAF comes in the relationship between the OV and SVs. A tactical organization chart may be thought of as the starting point for an OV while a network connectivity diagram may be thought of as the starting point for an SV.

Simplistically, the OVs and SVs establish *what* systems must connect, and the SVs and technical standards view establish *how* systems must connect. From an engineering perspective, OVs are representations of requirements. The SVs describe how those requirements are implemented. DoDAF-compliant architectures constructed with this symmetry in mind have traceability between systems and requirements.

We exploit this traceability by considering the relationship of security policy to validating information assurance architecture. A simple and effective rule for security design is the principle of least privilege. That is, allow only the minimum essential connectivity and functionality. This is a principle easier to enunciate than it is to implement. Detailed requirements are needed to answer the question, “What is the minimum required functionality and connectivity?” From an information assurance perspective, security policy translates operational requirements into system requirements. This, then, is the basis of the methodology to develop information assurance architecture from DoDAF-compliant architecture.

Developing Information Assurance Architecture

What are the security requirements for the system(s) of interest? Our starting point is a set of OVs, specifically the OV-2 – the Operational Node Connectivity Description [6]. The DoDAF Deskbook [7] gives a high-level example of showing security attributes to a node as shown in Figure 2. Our methodology goes into more detail.

We start with the OV-2 Operational Node Connectivity Description. These nodes are the entities that are represented in the architecture. An object-oriented modeler would consider these nodes to be actors. An operational planner would consider these nodes to be a tactical element. These nodes can be represented at various levels of detail. A brigade combat team could be represented as one node, as a collection of battalion nodes, as an even larger number of company nodes, or quite probably a collection of nodes at different levels of abstraction.

For information assurance architecture, it is necessary to list every single system. A 99 percent solution is not effective! So, the cardinality may vary; that is, one node may represent one system, or one node may represent many systems. In our example, we use one-to-one mapping because it is easier. However, it is quite feasible, for example, to represent a network operations center as a single node and then have multiple system nodes that are part of the operations center node.

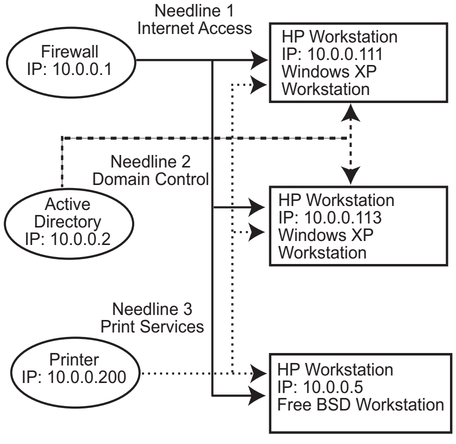
Consider the OV-2 Operational Node Connectivity Description of our Information Assurance Laboratory as shown in Figure 3.

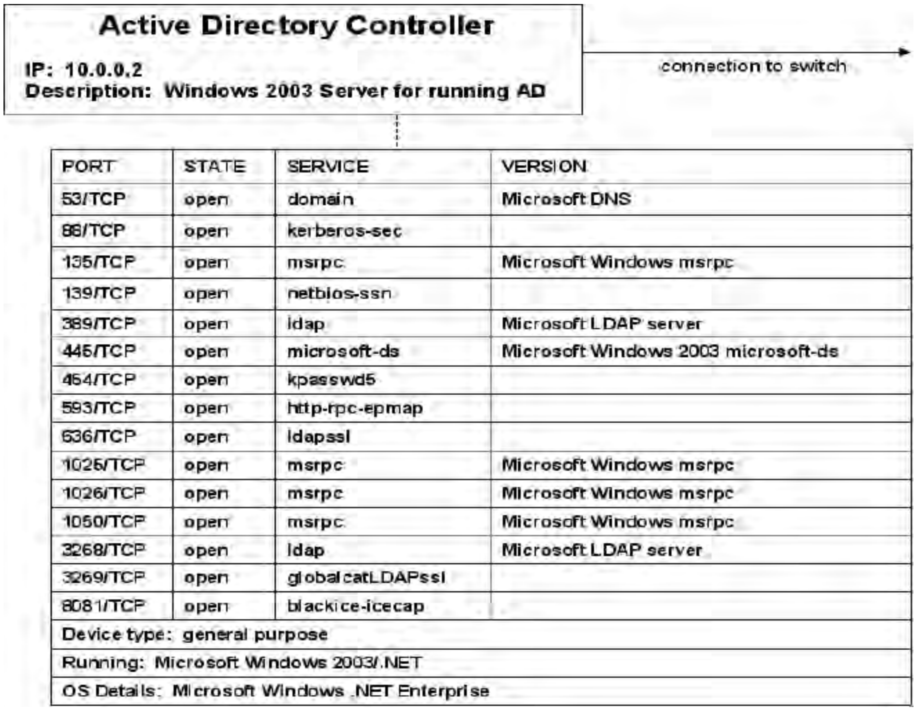
Here we have gone beyond the minimal DoDAF standard and provided some additional information on what each node does. We then look at activities that each node is required to perform. The OV-5 Operational Activity Model [6] captures the required activities for each node. Now, it is a standard practice to go from the OVs to construct the SVs in new acquisitions. (In constructing *as-is architecture*, i.e., documenting existing systems, it is common to construct the SVs first.) To construct useful information assurance architecture, it is necessary to drill down to a greater level of detail. To achieve this, we base the security policy on the OVs. It could be argued that, in lieu of a separate security policy, the policy requirements be enumerated in an OV-6A Operational Rules Model [6] – we merely call it out separately from a DoDAF product for use as an operational security policy.

The construction of the security policy is oriented to the OV-2 and OV-5. For example, to allow file transfer protocol (FTP) access, we determine which nodes require activities that need FTP as opposed to some other, more secure transfer protocol. DoDAF traceability requires a consistent numbering policy. Our security policy representation is numerically indexed to the nodes and activities in the OVs. Since it is a DoDAF requirement that each system in the SVs (specifically the SV-1) be tied to a node in the OV-2, we now have a security crosswalk between the OVs (requirements) and the SVs (implementation).

Finally, for each system in SV-1, we list

Figure 3: *Simple OV-2 of the Auburn Information Assurance Laboratory*


Needline 1 Internet Access
Firewall IP: 10.0.0.1
HP Workstation IP: 10.0.0.111 Windows XP Workstation
Needline 2 Domain Control
Active Directory IP: 10.0.0.2
HP Workstation IP: 10.0.0.113 Windows XP Workstation
Needline 3 Print Services
Printer IP: 10.0.0.200
HP Workstation IP: 10.0.0.5 Free BSD Workstation



Active Directory Controller

IP: 10.0.0.2
Description: Windows 2003 Server for running AD

connection to switch

PORT	STATE	SERVICE	VERSION
53/TCP	open	domain	Microsoft DNS
88/TCP	open	kerberos-sec	
135/TCP	open	msrpc	Microsoft Windows msrpc
139/TCP	open	netbios-ssn	
389/TCP	open	ldap	Microsoft LDAP server
445/TCP	open	microsoft-ds	Microsoft Windows 2003 microsoft-ds
464/TCP	open	kpasswd5	
593/TCP	open	http-rpc-epmap	
636/TCP	open	ldapssl	
1025/TCP	open	msrpc	Microsoft Windows msrpc
1026/TCP	open	msrpc	Microsoft Windows msrpc
1050/TCP	open	msrpc	Microsoft Windows msrpc
3268/TCP	open	ldap	Microsoft LDAP server
3269/TCP	open	globalcatLDAPssl	
8081/TCP	open	blackice-icecap	
Device type: general purpose			
Running: Microsoft Windows 2003/.NET			
OS Details: Microsoft Windows .NET Enterprise			

Figure 4: *Partial SV-1 With Information Assurance Details (Active Directory Controller)*

the minimal set of required services, processes, and open ports as shown in Figure 4, which shows a partial SV-1 derived from the OV-2 Active Directory Node in Figure 3. Based on the relationship between the OVs, security policy, and SVs, we now have a holistic information assurance architecture that can be verified on each system and validated against requirements.

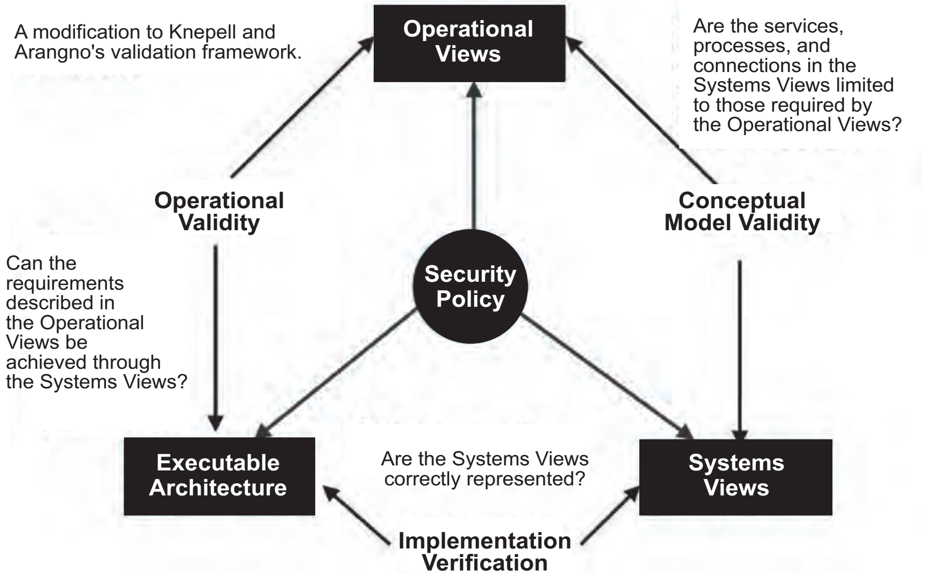
Verification and Validation of Information Assurance Architecture

Since the current Defense Acquisition System mandates using the DoDAF, developers have strong motivation to demonstrate that their architectures are valid and internally consistent. This holistic approach to system specification and connectivity is greatly useful in designing, verifying, and validating information assurance architecture. This relationship is shown in Figure 5.

OVs are fully defined in Volume 2 of the DoDAF [6]. Succinctly, OVs are representations of requirements. Consequently, there is a direct relationship between OVs and SVs. Figure 5 is a variation of Knepell and Arangno's validation structure adapted for information assurance application of the DoDAF [8].

An operational concept is not valid if it cannot be supported by the systems available in theater. So in this sense, the SVs validate the conceptual model of the OV as shown in Figure 5. Conversely, the validity of systems architecture can be evaluated against how well it supports the requirements documented in the operational architecture.

Figure 5: *Validating DoDAF-Based Information Assurance Architecture*



A modification to Knepell and Arangno's validation framework.
Operational Views
Are the services, processes, and connections in the Systems Views limited to those required by the Operational Views?
Operational Validity
Conceptual Model Validity
Security Policy
Can the requirements described in the Operational Views be achieved through the Systems Views?
Executable Architecture
Are the Systems Views correctly represented?
Systems Views
Implementation Verification

The employment of executable architecture adds a new and needed dimension to the verification and validation of DoDAF-compliant information assurance architecture. Executable architectures can assess the validity of an operational concept. While the SVs may provide the needed connectivity to support the operational concept described in the OVs, the SVs alone do not give sufficient insight into meeting operational performance and capacity needs. It can be argued that required performance can be extrapolated from the SVs, but executable architecture can provide a much more dynamic and flexible means of evaluation.

From an information assurance viewpoint, executable architectures can evaluate network and system design in terms of resistance and resiliency in the face of denial-of-service attacks [9]. A thorough discussion of executable architectures is beyond the scope of this article, but the heart of executable architecture is a network simulation constructed from the systems detailed in the DoDAF SVs and exercised by applying dynamic behavior across the system connections required in the OVs. Executable architectures are described in detail in [5].

Central to this validation structure is the security policy that is derived from the OVs, enforced in the SVs, and must be modeled in the executable architecture.

Conclusion

The DoDAF is mandated for use in the DoD Acquisition System – compliant architectures are a significant investment, and it makes sense to leverage this investment rather than undertaking a costly independent information assurance effort. We have briefly illustrated some specific ways to implement information assurance architecture and how to verify and validate such architecture.◆

References

1. Department of Defense. "DoD Instruction 5000.2." Operation of the Defense Acquisition System. Washington, D.C.: DoD, 12 May 2003: 2.
2. Hamilton Jr., J.A. "Why Programming Languages Matter." CrossTalk Dec. 1997 <www.stsc.hill.af.mil/crosstalk/frames.asp?uri=1997/12/languages.asp>.
3. Department of Defense. "Chairman,

Joint Chiefs of Staff Instruction 3170.01E.” Joint Capabilities Integration and Development System. Washington, D.C.: DoD, 11 May 2005.

4. Department of Defense. DoD Architecture Framework Vers. 1.0 Vol. I: Definitions and Guidelines. Washington, D.C.: DoD, 9 Feb. 2004: 1-2 <www.defenselink.mil/nii/doc>.
5. Hamilton Jr., J.A. Modeling Command and Control Interoperability: Cutting the Gordian Knot. San Diego, CA: SCS Press, 2004: 85-99 <www.eng.auburn.edu/users/hamilton/security/spawar>.
6. Department of Defense. DoD Architecture Framework Vers. 1.0 Vol. II: Product Descriptions. Washington, D.C.: DoD, 9 Feb. 2004 <www.defenselink.mil/nii/doc>.
7. Department of Defense. DoD Architecture Framework Vers. 1.0 Deskbook. Washington, D.C.: DoD, 9 Feb. 2004: 2-79 <www.defenselink.mil/nii/doc>.
8. Knepell, P.L., and D.C. Arangno. Simulation Validation. Los Alamitos, CA: IEEE Computer Society Press, 1993.
9. Chatam, J.W. Using Strategic Firewall Placement to Mitigate the Effects of Distributed Denial of Service Attacks. Thesis. Auburn University, Aug. 2004.

About the Author

John A. “Drew” Hamilton Jr., Ph.D., is an associate professor of computer science and software engineering at Auburn University and director of Auburn University’s Information Assurance Laboratory. Prior to his retirement from the U.S. Army, he served as the first director of the Joint Forces Program Office and on the staff and faculty of the U.S. Military Academy, as well as chief of the Ada Joint Program Office. He has a Bachelor of Arts in journalism from Texas Tech University, master’s degrees in systems management from the University of Southern California and in computer science from Vanderbilt University, and a doctorate in computer science from Texas A&M University.

Auburn University
107 Dunstan Hall
Auburn, AL 36849
Phone: (334) 844-6360
Fax: (334) 844-6329
E-mail: hamilton@eng.auburn.edu

Web Sites

Air Force Research Laboratory

www.afrl.af.mil

The United States Air Force Research Laboratory (AFRL) leads the discovery, development, and integration of affordable warfighting technologies. The AFRL is a full-spectrum laboratory of approximately 9,500 people, responsible for planning and executing the Air Force’s entire science and technology budget of nearly $1.7 billion, including basic research, applied research, and advanced technology development. AFRL partners include universities and industry with whom the AFRL invests almost 80 percent of its budget; customers include the Air Force major commands, which operate and maintain the Air Force’s weapon systems.

National Aeronautics and Space Administration

www.nasa.gov

The National Aeronautics and Space Administration (NASA) conducts its work in four principle organizations, called mission directorates: aeronautics, exploration systems, science, and flight support. Closer to home, NASA’s aeronautics team is working with other government organizations, universities, and industry to fundamentally improve the air transportation experience.

Practical Software and Systems Measurement Support Center

www.psmsc.com

The Practical Software and Systems Measurement (PSM) Support Center is sponsored by the Department of Defense (DoD) and the U.S. Army. It provides project managers with the objective information needed to successfully meet cost, schedule, and technical objectives on programs. PSM is based on actual measurement experience with DoD, government, and industry programs. The Web site also has the most current version of the *PSM Guidebook*.

Coming Events

March 6-9
Software Engineering Process Group (SEPG) 2006
Nashville, TN
www.sei.cmu.edu/sepg

March 13-15
International Symposium on Secure Software Engineering
Washington, D.C.
www.jmu.edu/iiia/issse

March 15-16
International Conference on Information Warfare and Security
Princess Anne, MD
http://academic-conferences.org/iciw/iciw2006/iciw06-home.htm

March 20-22
Association for Configuration and Data Management 11th Annual Technical and Training Conference
Sparks, NV
www.acdm.org/2006/conference.php

April 2-6
9th Communications and Networking Simulation Symposium
Huntsville, AL
www.scs.org/confernc/springsim/springsim06/cfp/cns06.htm

April 3-7
The 3rd International Conference on Software Process Improvement
Orlando, FL
www.icspi.com

April 10-12
3rd International Conference on Information Technology: New Generations
Las Vegas, NV
www.itng.info

May 1-4
2006 Systems and Software Technology Conference
STC Systems & Software Technology Conference
Salt Lake City, UT
www.stc-online.org